The Effect of Using Pitch and Duration for Symbolic Music Retrieval

Iman S. H. Suyoto and Alexandra L. Uitdenbogerd
School of Computer Science and Information Technology
RMIT University
Vic. 3001 Australia
{*Iman.Suyoto,Alexandra.Uitdenbogerd*}*@rmit.edu.au*

Abstract *Quite reasonable retrieval effectiveness is achieved for retrieving polyphonic (multiple notes at once) music that is symbolically encoded via melody queries, using relatively simple pattern matching techniques based on pitch sequences. Earlier work showed that adding duration information was not particularly helpful for improving retrieval effectiveness. In this paper we demonstrate that defining the duration information as the time interval between consecutive notes does lead to more effective retrieval when combined with pitch-based pattern matching in our collection of over 14 000 MIDI files.*

Keywords Music information retrieval, Information retrieval, Multimedia resource discovery, Pattern matching

1 Introduction

The field of music information retrieval has as its aim the development of technology to enable users to find music that they are searching for. There are many ways that users may wish to search for music, such as locating information about a song for which a small fragment is remembered, finding music that is of a similar style to an example, or simply searching for music that the user might like. The reason for searching could be simply to satisfy the user's curiosity, check for copyright infringement, or to purchase new music.

One of the main problems studied in the field of music information retrieval is that of retrieving music given a query that is a melody fragment, such as a few notes of the sung component of a verse of a song. The problem's complexity varies depending on the format of the query and the music collection, with the simplest being search of a symbolically encoded collection of melodies using a symbolically encoded melody. In this paper, we use symbolic melody queries and a polyphonic (multiple notes at once) collection of music. Most of our early work [29, 34, 35] was restricted to search using a representation of both queries and music from the collection as sequences of pitches. Rhythm information was ignored. This approach was shown to be competitive with more complex techniques in recent evaluation exchanges [24, 31, 32] where the collection was symbolically encoded. However, improvement may be possible with the introduction of rhythm information, potentially allowing matching techniques to yield greater effectiveness for sung queries that are likely to be less precise than those issued via a musical keyboard or text-based encoding. In our experiments we explore two different methods of encoding rhythm: encoding the duration of each note in a melody and *inter-onset intervals* (IOI) — the time interval between successive notes. We found that improvement in retrieval effectiveness is possible using an IOI representation of rhythm.

Proceedings of the 13th Australasian Document Computing Symposium, Hobart, Australia, 8 December 2008.

2 Related Work

Much of previous research has shown that the pitch feature is sufficient to support effective content-based retrieval of music. The usage of both pitch and rhythm has also been examined in past work by, for example, McNab et al. [15], Chen and Chen [1], Lemström et al. [12], Dannenberg et al. [2], Ferraro and Hanna [5], Hanna et al. [8], Typke et al. [27], and Lemström et al. [13]. Other than in our previous work [23], the relative value of these features for matching on large polyphonic collections has not been measured. In addition, the benefit of string-matching approaches in this scenario have not been thoroughly investigated yet. We discuss each of these papers below.

McNab et al. [15] investigated what combination of pitch and duration features has the best discriminatory power to distinguish one musical piece from others. Their collection consisted of 9 600 folksong melodies. They examined both exact matching and approximate matching (using dynamic programming as given in Mongeau and Sankoff [18]). To represent the pitch component of notes, they used pitch interval, which is the difference in pitch between two adjacent notes, and pitch contour, which is the movement direction from a previous note to a current note, described further in Uitdenbogerd and Yap [33]. They found that for highly effective exact matching with rhythm, five notes are sufficient. Without rhythm, about seven notes are

required. For approximate matching (with rhythm), the number of required notes increases to twelve.

A technique for retrieval by rhythm was proposed in Chen and Chen [1]. Every piece was represented by a rhythm string, representing solely the rhythmic patterns in that piece. In particular, a piece was divided into measures, and the note durations in every measure in a piece were captured as a unit. Pitches were ignored. Every measure was stored as a node in a tree-based index structure. Their paper emphasises the efficiency of their approach, but fails to present how effective it is. Their test collection only consisted of 102 folk songs (the format of which is unspecified). The relatively small size of the collection and the lack of effectiveness benchmark make the merit of this approach questionable.

Lemström et al. [12] introduced a technique that represents a note as a combination of its pitch interval (with respect to the note preceeding itself) and duration of a monophonic music sequence, called relative interval slope. A sequence consists of n notes, each of which is a pair of its pitch and its duration. The interval slope sequence consists of n symbols, each is the signed difference between the pitch of the current note and that of the previous note, over the duration of the previous note. The first symbol is a special case; it is the pitch of the first note over the the duration of the last note. If every symbol in the interval slope sequence is denoted by $a_i; 1 \leq i \leq n$, the relative interval sequence consists of n symbols, each is a_i for $1 \leq i \leq 2$ or $\frac{a_i}{a_{i-1}}$ for $i > 2$. They conducted their experiment on a collection of 6 070 monophonic MIDI tracks. Only exact matches were considered. It is not clear how many queries were used. It is mentioned that the experiment run consisted of 18 000 searches, but the number of unique queries is not mentioned. For queries with pattern length of 13 up to 20, no false positive was generated.

In Dannenberg et al. [2], rhythmic information was used for query-by-humming retrieval, with an answer collection of MIDI files. Three melody encoding approaches were evaluated. In the first approach, a note is represented using its pitch interval and inter-onset interval ratio. An inter-onset interval ratio is encoded as a quantised value of five possible values as devised in Pardo and Birmingham [19], which a pitch interval is encoded as a quantised value of 25 possible values. These make this encoding tempo-invariant and transposition-invariant. Edit distance was used as the similarity measure. In the second approach, based on Mazzoni and Dannenberg [14], a piece was divided into frames of equal time length, from each of which the fundamental frequency is estimated. In this case, note boundaries were ignored. The obtained melody was then transposed 24 times, half a semitone each time. Dynamic time warping was used for matching. In the third approach, based on Meek and Birmingham [16], a note was represented using its pitch class and inter-onset interval, quantised based on a log scale. Matching was performed using a hidden Markov model. Two experiments were conducted. The first experiment involved 160 queries (80 for training and 80 for testing) and a collection of 10 000 synthetically generated pieces with a mean length of 40 notes as noise and 10 folk songs as targets. How the 10 000 pieces were generated is not described. As the result of this experiment, the third approach caused 73.75% of the test queries to obtain the target answer in the first rank position, but the results for the other two approaches were not reported. The second experiment used two query sets. The first query set consists of 131 queries, whereas the second one consists of 165 queries. The first query set was run against a collection of 258 Beatles pieces, and the second query set was run against a collection of 868 popular songs. The third approach was superior for the first query set, yielding a mean reciprocal rank value of 27.0% (compared to 21.0% for the second approach and 13.4% for the first approach). For the second query set, the second approach was superior, yielding a mean reciprocal rank value of 32.9% (compared to 31.0% for the third approach and 28.2% for the first approach).

Ferraro and Hanna [5] and Hanna et al. [8] explored the use of duration information for monophonic music matching. They examined using duration differences between two notes. It is not clearly specified which two notes are meant. Combination of similarity evidence is used to combine the pitch similarity score (s_{pitch}) with the duration similarity score (s_{duration}) using the formula given in Mongeau and Sankoff [18]: $s_{\text{total}} = s_{\text{pitch}} + k s_{\text{duration}}$ where k is a weighting parameter. They claim that at $k = 0.20$, using duration information improves retrieval effectiveness over the use of pitch only.[1] The statistical significance of their result is not reported. Ferraro and Hanna [5] and Hanna et al. [8] claim to obtain significantly different results from using duration and disagree with our conclusion [23] that says otherwise. However, they were using monophonic music, whereas our experiments used polyphonic music. On the improvement significance aspect, we admitted that there was a slight improvement when duration information was used, albeit not statistically significant. On the other hand, they have shown no proof of statistical significance of their claim. Moreover, they did not contrast the input sizes used in both papers. Their work used the testbed of MIREX 2005, which had a collection of 558 MIDI pieces with only 11 queries. This is clearly much smaller than ours (more than 10,000 pieces in the collection and 24 queries) and an indication that the complexity of the problem they were discussing was much smaller.

All work mentioned above involved the use of duration on monophonic collections. There has been research that attempts to use duration-based information on polyphonic music, such as Typke et al. [27] and

[1] The k value is reported in Hanna et al. [8] but not in Ferraro and Hanna [5].

Lemström et al. [13]. Typke et al. [27] described several retrieval tasks in MIREX 2006.[2] Two of them involved polyphonic music:

1. Symbolic melodic similarity using 1 000 polyphonic karaoke files with five queries (referred as the karaoke task herethereafter).

2. Symbolic melodic similarity using 10 000 MIDI files downloaded from the Web, most of which are polyphonic, with six queries (referred as the mixed polyphonic task herethereafter).

In their approach, a melody extraction routine was applied to obtain monophonic representations of the polyphonic pieces. A skyline algorithm[3] was used. Which specific skyline algorithm was not specified. These monophonic representations are divided into overlapping segments with different lengths. They used lengths of 5 to 16, except for the second task, where they used 5 to 7. The segments were then indexed using vantage indexing [36] using the Proportional Transportation Distance [28] as the distance measure. A note was represented as a two-dimensional point [28], with pitch and onset time as the dimensions. The duration of the note was used as the weight of the point. For the two tasks, their method achieved a MAP value of 0.875 and 0.903 respectively.

In Lemström et al. [13], a geometric sweepline algorithm called P3 was used. Every piece was represented by its piano roll [20] representation. The features used were pitch and the start and end times (which can be used to derive durations) of notes. To determine the similarity between a query and an answer, the maximum overlap was determined over keys to ensure transposition invariance. Although this caters for difference in keys, it will likely fail if the tempi of the query and the answer are different. To address this, they proposed SCALEDP3, which extends P3 by scaling the query tempo by a scaling factor. However, it performed poorly on the MIREX 2006 symbolic polyphonic retrieval tasks.

3 Feature Extraction

Our approach assumes that we are working with polyphonic symbolic music. The string representations mentioned in this paper imply that a sequence is one-dimensional, since we cannot have any overlap in a string. However, in polyphonic music, notes can overlap, and as such, it is two-dimensional. Previously, Uitdenbogerd and Zobel [29] showed that reducing the two-dimensional space into one dimension by extracting a representative note for a particular time point can support effective retrieval. The output from feeding polyphonic music into this process is therefore a monophonic melody, representing the polyphonic music. The ALL-MONO algorithm has been shown to be a highly effective melody extraction algorithm. If there is a note m of length l_m sounding at time t_m and another note n sounding at t_n so that $l_m + t_m > t_n$, then l_m will become $l'_m \leftarrow t_n - t_m$. In other words, note overlaps are removed. The ALL-MONO algorithm is outlined in Algorithm 1.

Algorithm 1 ALL-MONO melody extraction algorithm. A note is expressed as a tuple $n = \langle p, d, o \rangle$ where p is the pitch, d is the duration, and o is the onset time. The base index is 0. P is the sequence of the representative bass part. "π_x" is the relational operator for projecting the x attribute.

Require: array of notes **N**
  Sort **N** by ascending onset time as the first sort key and descending pitch as the second sort key.
  {Start taking the highest note at any onset time.}
  for $i = 0 \ldots |\mathbf{N}| - 2$ **do**
    if $(\pi_o n_i \neq \pi_o n_{i+1})$ **then**
      Append $\pi_p n_i$ to P.
    end if
    if $(\pi_o n_i + \pi_d n_i > \pi_o n_{i+1})$ **then**
      $d' \leftarrow \pi_o n_{i+1} - \pi_o n_i$
      $n_i \leftarrow \langle \pi_p n_i, d', \pi_o n_i \rangle$
    end if
  end for
  Append $\pi_p n_{|\mathbf{N}|-1}$ to P.
  {End.}
  return P

4 Matching Technique

To support approximate matching, we convert the melody into standardisations. The pitch standardisation used for the experiments described in this paper is the directed modulo-12 approach [23, 26, 30], described in Section 4.1. As our experiments also make use of the duration feature in notes, we also need to encode the durations into a searchable representation. For this purpose, we use the extended contour standardisation, to be described in Section 4.2.

4.1 Pitch Directed Modulo-12 Standardisation

In the directed modulo-12 standardisation, a note is represented as a value r which is the interval between a note and its previous note scaled to a maximum of one octave [21, 30]:

$$r \equiv d(1 + ((I - 1) \bmod 12)) \qquad (1)$$

where I is the interval between a note and its previous note (absolute value) and d is 1 if the previous note is lower than the current note, -1 if higher, and 0 if otherwise. For example, the melody shown in Fig. 1 is encoded as "`7 4 1 -5 -5 2 3 -2 -1 -2`".[4]

[2] See `http://www.music-ir.org/mirex2006`.

[3] A skyline algorithm takes from a set of overlapping items the one with the extreme value of a certain feature of set of features. The ALL-MONO algorithm (Algorithm 1) is an example skyline algorithm.

[4] A figure is treated as a symbol. Hence, it is a 10-symbol string.

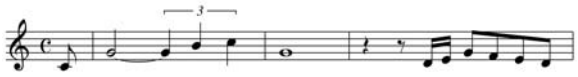

Figure 1: "Melbourne Still Shines" by ade ishs.

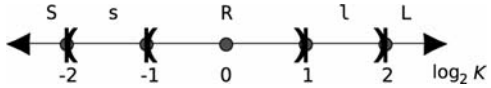

Figure 2: Duration extended contour quantisation. $K = \lambda_C/\lambda_P$ where λ_C and λ_P are respectively the current and previous note durations. The current note is represented as "R" if $|\log_2 K| < 1$; "l" if $1 \leq \log_2 K < 2$; "L" if $\log_2 K \geq 2$; "s" if $-2 < \log_2 K \leq -1$; and "S" if $\log_2 K \leq 2$.

4.2 Duration Extended Contour Standardisation

The extended contour standardisation is partly inspired by a pitch standardisation called the pitch extended contour standardisation [30], which encodes a note as a movement direction of the previous note pitch to its pitch. There are five distinct symbols, each representing a set of pitch intervals: "S" if the current note is the same pitch as the previous note, "u" if the current note pitch is a little higher than the previous note pitch, "U" if the current note pitch is much higher than the previous note pitch, "d" if the current note pitch is a little lower than the previous note pitch, and "D" if the current note pitch is much lower than the previous note pitch.

Just as in pitch contour-based standardisations, the extended contour standardisations also employ five distinct symbols to represent a note. In the case of duration, we use "S", "s", "R", "l", and "L" for "much shorter", "a little shorter", "same", "a little longer", and "much longer" respectively. Interestingly, Moles [17] describes an approach for encoding duration quantisation. The quantisation we use in our experiments is based on the encoding given in that literature. Let λ_C be the current note, λ_P be the previous one, and $K = \lambda_C/\lambda_P$. A note is represented based on the ranges of $\log_2 K$ as illustrated in Figure 2. For example, the melody shown in Figure 1 is represented as "L S R L S R l R R R".

4.3 Alignment

Kageyama et al. [11] suggested the use of note durations as penalty scores for insertion and deletion operations in calculating weighted edit distances. How the scores are calculated is not formally defined however. In this work, we also use a dynamic programming technique, that is, the local alignment algorithm [7], It is useful to find the substring with the highest similarity within a string. Query tunes are usually represented by short strings while answer tunes are usually represented by long strings, so the alignment is more suitable than global alignment [29].

For a query-answer pair, two scores are produced: one pitch similarity score, and one duration similarity score. These scores are to be fused using a similarity evidence combination technique described in the following section.

4.4 Combining Pitch and Duration Similarity Scores

We experiment with a vector model to combine similarity evidence from both pitch and duration matching. The pitches and durations are represented using the respective standardisations. For the purpose of fusing the pitch and duration similarity scores, they are modelled as vectors perpendicular to each other, making the resultant similarity vector become the overall similarity. The following formula is based on one in our previous work [22], where we represent pitch and duration as perpendicular unit vectors. To allow better fine-tuning, we now also assign weights for both pitch and duration components:

$$\vec{\Sigma} \equiv w_\pi \varsigma_\pi \hat{\pi} + w_\delta \varsigma_\delta \hat{\delta} \tag{2}$$

where $\vec{\Sigma}$ is the resultant similarity vector, ς_π is the pitch similarity, ς_δ is the duration similarity, w_π and w_δ are both weight constants, and $\hat{\pi}$ and $\hat{\delta}$ are respectively pitch and duration unit vectors. Ranking is then based on the magnitude of the resultant similarity vector, $|\vec{\Sigma}| = \sqrt{w_\pi^2 \varsigma_\pi^2 + w_\delta^2 \varsigma_\delta^2}$.

5 Experimental Setup

As the aim of our experiment is to identify whether note duration information is useful for melody retrieval, we use a collection of polyphonic MIDI files and a set of queries manually constructed by human subjects. The collection contains 14 193 MIDI files, which form a superset of the collection used in experiments by Uitdenbogerd and Zobel [29, 34] and Uitdenbogerd et al. [35]. A total of 24 queries were constructed by a musician after listening to a set of polyphonic pieces. The relevance judgement set was generated by human users. They were presented with top answers from several matching techniques and asked to give a binary relevance judgement. More detail can be found in Uitdenbogerd et al. [35].

As the baseline of our experiment, for pitch matching, we used $M(x,x) = 1$ for a match, $M(x,y)|_{x \neq y} = -1$ for a mismatch, and $I = -2$ for an insertion/deletion (see Section 4.3) as used elsewhere [23, 29, 34]. For duration matching, we used 21 scoring matrices as in Suyoto and Uitdenbogerd [23]. The scoring matrices were obtained by varying the variables $a, b, c, \ldots, i$ shown in Figure 3, as detailed in Table 1. The matrix means if there is a match "S"-"S", $M(\text{"S"},\text{"S"}) = c$; a mismatch "S"-"s", $M(\text{"S"},\text{"s"}) = d$; etc. At any time, $a \geq b \geq c \geq d \geq e \geq f \geq g \geq h \geq i$. The values of these variables correspond to the rewards/penalties based on the likelihood that there is an actual match when the symbols do not actually match. In other

	S	s	R	l	L
S	c	d	f	h	i
s	d	b	e	g	h
R	f	e	a	e	f
l	i	g	e	b	d
L	h	i	f	d	c

Figure 3: Scoring matrix for duration extended contour standardisation. "S", "s", "R", "l", and "L" respectively indicate a "much shorter", an "a little shorter", a "same", an "a little longer", and a "much longer".

SS	a	b	c	d	e	f	g	h	i
1	1	1	1	1	−1	−1	−1	−1	−1
2	2	1	1	1	−1	−1	−1	−1	−1
3	3	1	1	1	−1	−1	−1	−1	−1
4	3	2	1	1	−1	−1	−1	−1	−1
5	3	3	1	1	−1	−1	−1	−1	−1
6	3	3	2	1	−1	−1	−1	−1	−1
7	3	3	3	1	−1	−1	−1	−1	−1
8	3	3	3	3	−1	−1	−1	−1	−1
9	3	3	3	2	−1	−1	−1	−1	−1
10	3	3	3	1	−1	−1	−1	−1	−1
11	3	3	3	0	−1	−1	−1	−1	−1
12	3	3	3	−1	−1	−1	−1	−1	−1
13	3	2	1	0	−2	−3	−3	−3	−3
14	3	2	1	0	−3	−3	−3	−3	−3
15	3	2	1	0	−1	−2	−3	−3	−3
17	3	2	1	0	−1	−1	−3	−3	−3
17	3	2	1	0	−1	−1	−2	−3	−3
18	3	2	1	0	−1	−1	−1	−3	−3
19	3	2	1	0	−1	−1	−1	−2	−3
20	3	2	1	0	−1	−1	−1	−1	−3
21	3	2	1	0	−1	−1	−1	−1	−2
22	3	2	1	0	−1	−1	−1	−1	−1

Table 1: Scoring schemes (SS) for duration extended contour standardisation. For all scoring schemes, $a \geq b \geq c \geq d \geq e \geq f \geq g \geq h \geq i$.

words, if a symbol is replaced by a substitute, the matrix values represent how much it will change the rhythmic pattern of the melody. If an "R" matches an "R" (thus the score a is rewarded), it is very likely that the two notes represented by the symbols have the same relative duration or inter-onset interval. By an extreme contrast, the likelihood that two notes, each represented by "S" and "L" (thus the score i is given), have the same relative duration or inter-onset interval is small.

6 Results

In our experiment, queries were matched against all tunes in our collection 23 times, once for pitch matching using the directed modulo-12 standardisation and 22 times for duration matching using the 22 scoring schemes.

To combine pitch and duration similarities using Equation 2, we used ten different w_π/w_δ values: ∞ and $0,1,2,\ldots,9$. The first one is the baseline performance, that is, duration information is ignored ($w_\delta = 0$). The

Baseline MAP value = 0.326.

w_π/w_δ	Scoring Scheme	
	1	2
0	0.016	0.019
1	0.143	0.060
2	0.285	0.240
3	0.339	0.276
4	0.346	0.289
5	0.353	0.332
6	0.353	0.338
7	0.353	0.340
8	0.353	0.341
9	0.353	0.346

Table 2: MAP values for various w_π/w_δ using durations. The best values for each w_π/w_δ are highlighted.

w_π/w_δ	MAP
10	0.353 106 549 666 292
11	0.353 106 844 200 076
12	0.353 107 176 261 353
	⋮
18	0.353 107 351 475 871
19	0.353 107 351 475 871
20	0.353 107 351 475 871

Table 3: MAP values for $10 \leq w_\pi/w_\delta \leq 20$.

baseline performance has a MAP value of 0.326. The results of using other w_π/w_δ values are shown in Table 2. Due to space limitation, we only show the results for the scoring schemes that achieve the highest MAP for at least a value of w_π/w_δ. It can be seen that scoring scheme 1 performs consistently better than the other scoring schemes for various values of w_π/w_δ.

The MAP values for 1 and $w_\pi/w_\delta \geq 5$ appear to be approaching an extreme. Therefore, we performed further experiments with $10 \leq w_\pi/w_\delta \leq 20$ and obtained the results shown in Table 3. To assist us determining up to which w_π/w_δ the MAP value keeps increasing, we use 15 figures behind decimal point. We can see that the MAP values with w_π/w_δ starting from 17 are unchanging. See Figure 4 for the plot of MAP values with scoring scheme 1.

The best obtained MAP value is thus far 0.353. This is slightly higher than the baseline value of 0.326. We analyse further whether the two means are significantly different using a paired t-test as has been done elsewhere [25, 26]. It is found that incorporating duration information using the vector model does *not* lead to significant performance gain ($p > 0.2$).

The best scoring method, scoring scheme 1, implies that the "l" is treated the same as "L", and "s" is treated the same as "S". This is evident as $b = c = d$ and $e = f = g = h = i$. Therefore, if we were to remove the distinction between "much longer" and "a little longer," and also "much shorter" and "a little shorter," we would obtain representations with three distinct symbols (alphabets). Thus, the entropy [37], or the minimum number of bits required to store a symbol, defined as:

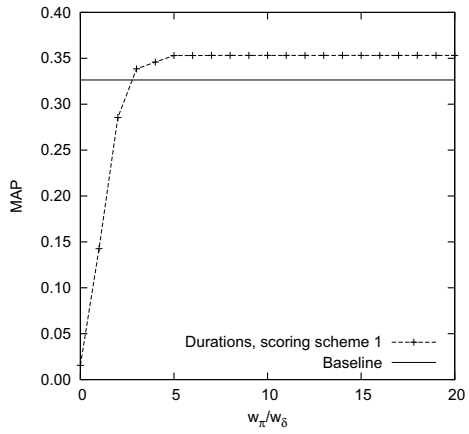

Figure 4: MAP values for pitch and duration matching using scoring scheme 1.

$$H = -\sum_{i=1}^{n} P(i)\log_2 P(i) \tag{3}$$

where $P(i)$ is the probability that the symbol i occurs, would be lower. We performed an informetric analysis as in Downie [3, 4], except that the sequences in our collection were not segmented into n-grams as our experiment assumed a unigram model. With the five-alphabet rhythm standardisation, the entropy of our whole collection is 1.858. With the three-alphabet rhtyhm standardisation, the entropy decreases to 1.491. A decrease in entropy also implies a decrease in information. However, our result shows that with less entropy, the effectiveness of retrieval increases. While Downie [3] believed that a higher information content of n-grams should cause retrieval performance to be better, our informetric analysis of our collection with a unigram model suggests that entropy itself may not be sufficient as an informetric analysis measure of likelihood that target pieces will be ranked higher (that is, high effectiveness). However, we are not certain whether Downie was referring to effectiveness or efficiency. The context hints that it was efficiency. What other measures should be used for effectiveness remains an open question.

We have shown that with the method we propose, duration information does not significantly improve retrieval performance. However, as we shall see shortly, using inter-onset intervals yields a different outcome.

7 Using Inter-Onset Intervals

One advantage of using inter-onset intervals compared to durations is that inter-onset intervals are less susceptible to variations in articulations and are more sensitive to rhythmic variations. As an illustration, let us suppose that we have three melodic fragments as shown in Figure 5.

Our point of interest is the second and third notes. Using durations, the extended duration contour standardisation is "SL" for the three cases. In other words, rhythmic pattern differences are not captured.

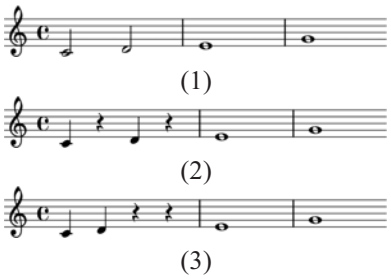

Figure 5: Melodic fragments with different note durations.

Algorithm 2 ALL-MONO-IOI melody extraction algorithm. A note is expressed as a tuple $n = \langle p, d, o \rangle$ where p is the pitch, d is the duration, and o is the onset time. The base index is 0. P is the sequence of the representative bass part. "π_x" is the relational operator for projecting the x attribute.

Require: array of notes **N**
  Sort **N** by ascending onset time as the first sort key and descending pitch as the second sort key.
  {Start taking the highest note at any onset time.}
  for $i = 0 \ldots |\mathbf{N}| - 2$ **do**
    if $(\pi_o n_i \neq \pi_o n_{i+1})$ **then**
      Append $\pi_p n_i$ to P.
    end if
    $d' \leftarrow \pi_o n_{i+1} - \pi_o n_i$
    $n_i \leftarrow \langle \pi_p n_i, d', \pi_o n_i \rangle$
  end for
  Append $\pi_p n_{|\mathbf{N}|-1}$ to P.
  {End.}
  return P

Using inter-onset intervals, the extended duration contour standardisation is "SL" for the first and second cases, and "Ll" for the third case. The difference between the first and second melodies is the articulation of the notes in the first bar, yet they both have the same rhythmic pattern. The difference is successfully picked up by inter-onset intervals. A musically-trained user is less likely to make rhythmic pattern errors when issuing queries. Articulation differences are less often considered as errors. Therefore, inter-onset intervals are more likely to be viable to improve retrieval effectiveness.

We modified the ALL-MONO algorithm so that the durations of a note is replaced by the time interval between itself and the following note. This is done indiscriminatively on the highest note at all onset times (excluding, the last note). Therefore, the difference between ALL-MONO and this algorithm (called ALL-MONO-IOI herethereafter) is that in ALL-MONO-IOI, there is no check whether the time to finish playing a note is after its following note. ALL-MONO-IOI is given as Algorithm 2.

Using ALL-MONO-IOI, we obtained a new set of duration-based representations of the pieces in our query set and collection. We used the same experimental setup outlined in Section 5, with this new

Baseline MAP value = 0.326.

w_π/w_δ	Scoring Scheme			
	1	2	12	13
0	0.025	0.017	0.035	0.022
1	0.176	0.130	0.051	0.051
2	0.322	0.236	0.156	0.207
3	0.318	0.271	0.232	0.281
4	0.320	0.318	0.262	0.307
5	0.319	0.324	0.314	0.313
6	0.319	0.327	0.327	0.353
7	0.319	0.327	0.346	0.356
8	0.319	0.327	0.348	0.356
9	0.319	0.327	0.348	0.356

Table 4: MAP values for various w_π/w_δ using inter-onset intervals. The best values for each w_π/w_δ are highlighted.

w_π/w_δ	**MAP**
10	0.355 544 269 543 587
11	0.355 550 207 447 156
12	0.355 571 651 845 875
	⋮
38	0.355 704 894 395 940
39	0.355 704 894 395 940
40	0.355 704 894 395 940

Table 5: MAP values for $10 \leq w_\pi/w_\delta \leq 40$.

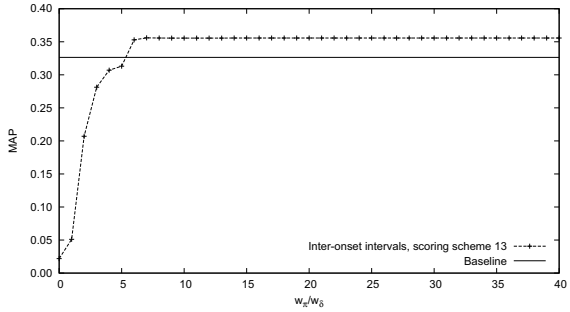

Figure 6: MAP values for pitch and inter-onset interval matching using scoring scheme 13.

set of representations. The MAP scores are given in Table 4.

The MAP values for 13 and $w_\pi/w_\delta \geq 5$ appear to be approaching an extreme. Therefore, we performed further experiments with $10 \leq w_\pi/w_\delta \leq 40$ and obtained the results shown in Table 5. To assist us determining whether there is an asymptotic value, we use 15 figures behind decimal point. We can see that the MAP values with w_π/w_δ starting from 38 are consistent. See Figure 6 for the plot of MAP values with scoring scheme 13.

The best obtained MAP value thus far is 0.356. This is slightly higher than the baseline value of 0.326. We analyse further whether the two means are significantly different using a paired t-test. It is found that incorporating inter-onset intervals using the vector model implies significant performance gain ($p < 0.05$).

8 Summary

We have compared two approaches of using duration-based information to improve retrieval effectiveness in this paper.

The first approach employs the durations of notes in the representative melody as extracted by the ALL-MONO algorithm [29]. Although the use of duration in addition to pitch improves retrieval effectiveness over the use of pitch only, the improvement is not significant. The second approach uses a modified version of ALL-MONO called ALL-MONO-IOI, which is similar to ALL-MONO except that the inter-onset intervals of representative melody notes are calculated. Although the modification is minor, our experimental setup shows that it has a significant impact on retrieval using duration-based information along with pitch. The retrieval effectiveness is improved significantly compared to using pitch only.

Acknowledgements We thank Falk Scholer and the anonymous reviewers for their input.

References

[1] J. C. C. Chen and A. L. P. Chen. Query by rhythm: An approach for song retrieval in music databases. In *Proceedings of IEEE International Workshop on Research Issues in Data Engineering*, pages 139–146, Feb. 1998.

[2] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo. The Musart testbed for query-by-humming evaluation. In Hoos and Bainbridge [9], pages 41–47.

[3] J. S. Downie. Informetrics and music information retrieval. In *Canadian Association for Information Science Proceedings of the 25rd Annual Conference*, pages 295–308. CAIS, June 1997.

[4] J. S. Downie. Informetrics and music information retrieval: An informetric examination of a folksong database. In *Canadian Association for Information Science Proceedings of the 26rd Annual Conference*. CAIS, June 1998.

[5] P. Ferraro and P. Hanna. Optimizations of local edition for evaluating similarity between monophonic musical sequences. In *Proceedings of Recherche d'Information Assistée par Ordinateur 2007*, Pittsburgh, USA, June 2007.

[6] M. Fingerhut, editor. *Proceedings of the Third International Conference on Music Information Retrieval*, Paris, France, Oct. 2002. IRCAM-Centre Pompidou.

[7] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK, 1997.

[8] P. Hanna, P. Ferraro, and M. Robine. On optimizing the editing algorithms for evaluating similarity between monophonic musical sequences. *Journal of New Music Research*, 36(4):267–279, Dec. 2007.

[9] H. H. Hoos and D. Bainbridge, editors. *Proceedings of the Fourth International Conference on Music Information Retrieval*, Baltimore, USA, Oct. 2003. Johns Hopkins University.

[10] International Music Information Retrieval Systems Evaluation Laboratory, editor. *Proceedings of the Second Annual Music Information Retrieval Evaluation eXchange*, Oct. 2006. URL `http://www.music-ir.org/mirex2006/`.

[11] T. Kageyama, K. Mochizuki, and Y. Takashima. Melody retrieval with humming. In *Proceedings of International Computer Music Conference 1993*, pages 349–351, 1993.

[12] K. Lemström, P. Laine, and S. Perttu. Using relative interval slope in music information retrieval. In *Proceedings of International Computer Music Conference 1999*, pages 317–320, Beijing, China, Oct. 1999.

[13] K. Lemström, N. Mikkilä, V. Mäkinen, and E. Ukkonen. Sweepline and recursive geometric algorithms for melodic similarity. In International Music Information Retrieval Systems Evaluation Laboratory [10]. URL `http://www.music-ir.org/mirex2006/`.

[14] D. Mazzoni and R. B. Dannenberg. Melody matching directly from audio. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the Second International Symposium on Music Information Retrieval*, pages 17–18, Bloomington, USA, Oct. 2001.

[15] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of ACM Digital Libraries 1996*, 1996.

[16] C. Meek and W. Birmingham. Johnny can't sing: A comprehensive error model for sung music queries. In Fingerhut [6], pages 124–132.

[17] A. Moles. *Information Theory and Esthetic Perception*. University of Illinois Press, Urbana, US, 1966.

[18] M. Mongeau and D. Sankoff. Comparison of musical sequences. In *Computers and the Humanities*, volume 24, pages 161–175. Kluwer, 1990.

[19] B. Pardo and W. Birmingham. Encoding timing information for musical query matching. In Fingerhut [6].

[20] L. Sitsky. *The Reproducing Piano Roll*. Department of Education, Canberra, Australia, Mar. 1979. ISBN 0-642-90543-6.

[21] I. S. H. Suyoto. Microtonal music information retrieval. Master's thesis, School of Computer Science and Information Technology, RMIT, Melbourne, Australia, 2003.

[22] I. S. H. Suyoto and A. L. Uitdenbogerd. Exploring microtonal matching. In C. L. Buyoli and R. Loureiro, editors, *Proceedings of the Fifth International Conference on Music Information Retrieval*, pages 224–231, Barcelona, Spain, Oct. 2004. Audiovisual Institute Pompeu Fabra University.

[23] I. S. H. Suyoto and A. L. Uitdenbogerd. Effectiveness of note duration information for music retrieval. In L. Zhou, B. C. Ooi, and X. Meng, editors, *Proceedings of the Tenth International Conference on Database Systems for Advanced Applications*, pages 265–275. Springer-Verlag, Apr. 2005. Published as LNCS 3453.

[24] I. S. H. Suyoto and A. L. Uitdenbogerd. Simple efficient n-gram indexing for effective melody retrieval. In International Music Information Retrieval Systems Evaluation Laboratory, editor, *Proceedings of the First Annual Music Information Retrieval Evaluation eXchange*, Sept. 2005. URL `http://www.music-ir.org/mirex2005/`.

[25] I. S. H. Suyoto, A. L. Uitdenbogerd, and F. Scholer. Effective retrieval of polyphonic audio with polyphonic symbolic queries. In J. Z. Wang, N. Boujemaa, A. Del Bimbo, and J. Li, editors, *Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 105–114, Augsburg, Germany, Sept. 2007.

[26] I. S. H. Suyoto, A. L. Uitdenbogerd, and F. Scholer. Searching musical audio using symbolic queries. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):372–381, Feb. 2008.

[27] R. Typke, F. Wiering, and R. C. Veltkamp. MIREX symbolic melodic similarity and query by singing/humming. In International Music Information Retrieval Systems Evaluation Laboratory [10]. URL `http://www.music-ir.org/mirex2006/`.

[28] R. Typke, F. Wiering, and R. C. Veltkamp. Transportation distances and human perception of melodic similarity. *ESCOM Musicae Scientiae*, (Discussion Forum 4A-2007):153–181, 2007.

[29] A. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In D. Bulterman, K. Jeffay, and H. J. Zhang, editors, *Proceedings of the 7th ACM International Conference on Multimedia '99*, pages 57–66, Orlando, USA, Nov. 1999. ACM Press.

[30] A. L. Uitdenbogerd. *Music Information Retrieval Technology*. PhD thesis, School of Computer Science and Information Technology, RMIT, Melbourne, Australia, 2002.

[31] A. L. Uitdenbogerd. Variations on local alignment for specific query types. In International Music Information Retrieval Systems Evaluation Laboratory [10]. URL `http://www.music-ir.org/mirex2006/`.

[32] A. L. Uitdenbogerd. N-gram pattern matching and dynamic programming for symbolic melody search. In International Music Information Retrieval Systems Evaluation Laboratory, editor, *Proceedings of the Third Annual Music Information Retrieval Evaluation eXchange*, Sept. 2007. URL `http://www.music-ir.org/mirex2007/`.

[33] A. L. Uitdenbogerd and Y. W. Yap. Was Parsons right? An experiment in usability of music representations for melody-based music retrieval. In Hoos and Bainbridge [9], pages 75–79.

[34] A. L. Uitdenbogerd and J. Zobel. Music ranking techniques evaluated. In M. Oudshoorn, editor, *Proceedings of the Twenty-Fifth Australasian Computer Science Conference*, pages 275–283, Melbourne, Australia, Jan. 2002.

[35] A. L. Uitdenbogerd, A. Chattaraj, and J. Zobel. Methodologies for evaluation of music retrieval systems. *INFORMS Journal of Computing*, 18(3):339–347, 2006. ISSN 1091-9856.

[36] R. H. van Leuken, R. C. Veltkamp, and R. Typke. Selecting vantage objects for similarity indexing. In Y. Y. Tang, P. Wang, G. Lorette, and D. S. Yeung, editors, *Proceedings of the 18th International Conference on Pattern Recognition*, pages 453–456, Hong Kong, China, Aug. 2006.

[37] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, San Fransisco, USA, second edition, 1999. ISBN 1-55860-570-3.